

# Scratch, Click & Vote

Mirosław Kutylowski, Filip Zagórski

Institute of Mathematics and Computer Science  
Wrocław University of Technology, Poland

End-to-end Voting Systems Workshop  
Washington DC, 13-14 X 2009

# Voter vs Election Authority

- ▶ Voter obtains a ballot from Election Authority
- ▶ How does voter know if her ballot is correctly encoded? – *randomized partial checking* or zero knowledge proof during pre-election audit
- ▶ How can one protect voter's privacy?  
Use ballot box (in SCV votes are cast through Proxy server)
- ▶ How one can assure that public data (commitments etc) does not reveal keys used for ballot-generation (covert-channel) – use *verifiable random function* or similar techniques

# Voter vs voting machine (PC) part I

- ▶ Machine cannot change voter's choice – voter obtains a receipt, which can be used to detect machine's misbehaviour.
- ▶ But at the same time, ballot and a receipt cannot be used to prove voter's choice
- ▶ Achieving these two properties is the hardest part in the system design.

# Voter vs PC part II

- ▶ If a machine has the same knowledge as a voter:
  - ▶ machine knows exactly how voter voted (privacy threat)
  - ▶ machine can change voter's choice (in some schemes)
  - ▶ online vote selling is possible
  - ▶ virus attacks are possible
- ▶ Solution: voter obtains additional “information” during registration (untappable channel) so:
  - ▶ PC learns voter's choice but does not know if vote will be counted (fakekey) [JCJ WPES05]
  - ▶ PC does not learn voter's choice [Chaum's SureVote, KZ IWSEC07, KZ SCV08]

# Voter vs PC consequences – usability

- ▶ PC learns voter's choice but does not know if vote will be counted (fakekey) [JCJ WPES05]
  - ▶ if voter votes ones – machine learns her choice
  - ▶ in fact voter is obliged to cast many (fake) votes to keep her choice secret
  - ▶ election with 3 runs with 1 out of 3 candidates each – 27 possibilities – vote 27 times (???)
- ▶ PC does not learn voter's choice [Chaum's SureVote, KZ IWSEC07, KZ SCV08]
  - ▶ SureVote – verifiability vs secrecy
  - ▶ KZ IWSEC07 – voter computes shift of the candidates
  - ▶ SCV – ThreeBallot-like vote casting

# Scratch, Click & Vote – ideas

- ▶ SCV is verifiable hybrid voting scheme:
  - registration ballots and encoders are delivered to voters by:  
traditional mail or email or physical visit in a  
registration office,
  - voting votes are cast over the Internet
- ▶ voter's computer is not trusted:
  - secrecy PC does not learn voter's choice
  - integrity PC cannot change voter's choice even into a  
random one
- ▶ receipt obtained by a voter does not prove voter's choice
  - masking ThreeBallot-like receipt
  - ambiguity voter may use many encoders

# Scratch, Click & Vote – ideas

- human verifiable:** a receipt obtained by a voter is human-readable and easy to examine by a moderately educated voter,
- voter friendly:** a voter (and her computer) needs not to perform any complicated (and hard to understand by an average voter) operations like: re-encryption, blind signatures etc.
- malware immune:** integrity of the elections and privacy of votes do not rely on any assumption on trustworthiness of the equipment used by the voter,
- efficient:** computational overhead as well as communication volume are low.

# Actors & vocabulary

## ► Actors:

**Election Authority** (EA) authority responsible for ballots preparation

**Proxy** authority responsible for preparation of encoders (simulates a ballot box)

**Registrar** authority responsible for the distribution of ballots and encoders

**Voter's PC** device used by a Voter

## ► Vocabulary:

**ballot** sheet of paper which a voter obtains from the Election Authority

**encoder** sheet of paper which a voter obtains from the Proxy, used to mask voter's choice from PC



# SCV – short scheme description

- V1 Start with straightforward Internet-version of the ThreeBallot (in fact “four-ballot”):
  - ▶ a voter visits *Proxy* webpage
  - ▶ Strauss’-like attacks on receipts
  - ▶  $2k + 1$  clicks in 1 out of  $k$  race & PC knows the choice!
- V2 Encoder (prepared by Proxy) is introduced:
  - ▶ exactly  $k$  clicks – every option gets exactly one click – PC does not know voter’s choice,
  - ▶ PC can change voter’s choice only with some probability,
  - ▶ but Proxy still knows voter’s choice
- V3 Ballots (prepared by EA) with permuted list of candidates:
  - ▶ confirmation codes – voter knows that vote is delivered
  - ▶ Proxy does not learn voter’s choice
  - ▶ EA does not learn who cast a vote (communicates directly with Proxy)

# Encoder

- ▶ Voter obtains a *ballot* from Election Authority
- ▶ Voter obtains many *encoders* from “Proxy” (many Proxies may be used)
- ▶ Voter lays them side by side

Candidate	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>
2 Jerry				
3 Edgar				
0 Ervin				
1 Donald				
$S_l$				

ballot (from EA)

n	Y	n	n	
n	Y	n	n	
Y	n	n	n	
n	n	n	Y	
$S_r$				

encoder (from Proxy)

Candidate	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>
2 Jerry	n	Y	n	n
3 Edgar	n	Y	n	n
0 Ervin	Y	n	n	n
1 Donald	n	n	n	Y
$S_l$	$S_r$			

ballot + encoder

- 

<i>Candidate</i>	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>
2 Jerry	n	Y	n	n
3 Edgar	n	Y	n	n
0 Ervin	Y	n	n	n
1 Donald	n	n	n	Y
$S_l$	$S_r$			



- ▶ Voter clicks on the screen on boxes which correspond to  $Y$  next to her candidate

PC screentransform (by Proxy)


# Vote casting

- ▶ Voter clicks on the screen on boxes which correspond to  $Y$  next to her candidate



ballot

<i>Candidate</i>	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>
2 Jerry	n	Y	n	n
3 Edgar	n	Y	n	n
0 Ervin	Y	n	n	n
1 Donald	n	n	n	Y
$S_l$	$S_r$			

PC screen

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

transform (by Proxy)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- ▶ Voter clicks on the screen on boxes which correspond to Y next to her candidate

PC screentransform (by Proxy)


# Vote casting

- 

<i>Candidate</i>	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>
2 Jerry	n	Y	n	n
3 Edgar	n	Y	n	n
0 Ervin	Y	n	n	n
1 Donald	n	n	n	Y
$S_l$	$S_r$			

PC screen

■			
		■	
■			
	■		

transform (by Proxy)


- ▶ Voter enters  $S_r$  (encoder serial number), proxy “translates” voter’s choice into FourBallot form

PC screentransform (by Proxy)

		×	×
×			×
	×	×	×
×		×	



- ▶ Voter enters  $S_i$  (ballot serial number), Proxy sends FourBallot form to the Election Authority

		×	×
×			×
	×	×	×
×		×	

$S_I$

# Vote casting

- ▶ Voter obtains as a receipt one of the FourBallot form ballots (oblivious transfer like protocol used)

ballot	transform (by Proxy)				receipt
<i>Candidate</i>	<i>R</i>	<i>S</i>	<i>T</i>	<i>U</i>	<i>T</i>
2 Jerry	n	Y	n	n	×
3 Edgar	n	Y	n	n	
0 Ervin	Y	n	n	n	×
1 Donald	n	n	n	Y	×
$S_l$	$S_r$				$t$

- ▶  $t = \text{sign}_{EA}(T, S_l)$  - confirmation token (like in Sure Vote)

# Security - PC/virus

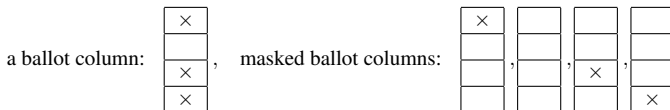
- ▶ Voter's PC can change voter's choice (with some probability):
  - ▶ PC does not know which row corresponds to the chosen candidate
  - ▶ modification can be detected by Proxy –  $\frac{1}{3k}$ , where  $k$  is the number of candidates
  - ▶ modification can be detected by voter – receipt ( $\frac{1}{4}$ )

# Security - Proxy, Election Authority

- ▶ Proxy can change voter's choice into a random one, but then a receipt will change - detection with probability  $\frac{1}{4}$
- ▶ Election Authority – negligible probability: Pre- and Post-election audits

# Security - other attacks

- ▶ There are known attacks on ThreeBallot (Strauss, Appeal):
  - ▶ FourBallots is much more immune – better probability distribution – Strauss' attack inefficient
  - ▶ moreover, it is easy to implement following modification (only electronic version) – instead of publishing every ballot, every ballot is split into masked ballots:



# SCV - Implementation

- ▶ Elections 8-10 VI 2009 [e-glosowanie.org](http://e-glosowanie.org), 6 500 voters
- ▶ Techniques used: Java, MySQL, PHP, Apache/Idea web servers, Solaris (EA), Red Hat (Proxy), Sun Cryptographic Accelerator (secret sharing, efficiency, admin passwords/master keys outside server's memory)
- ▶ See how it works (fully internet version – ballots are sent by email): [zagorski.im.pwr.wroc.pl/scv](http://zagorski.im.pwr.wroc.pl/scv)

# Summary - problems of Internet Voting

- ▶ Main problem of remote-voting systems is *physical coercion* (e. g. by the voter's spouse) but it is acceptable – mail-in voting.  
(Solution: well designed voter's registration)
- ▶ Why we do we really afraid of internet voting?:
  - ▶ possibility of massive undetectable fraud (malware on voter's PC)
  - ▶ possibility of massive online vote-selling (sell-your-vote software)

SCV is immune against both!

- ▶ Thank you for your attention